**ЛАБОРАТОРНА РОБОТА № 4**

**АВТОРИЗАЦІЯ**

Створити запит, який здійснює вихід користувача з усіх пристроїв.

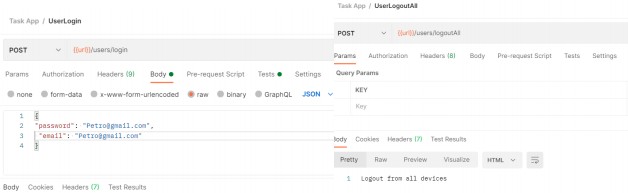
1. Створити POST /users/logoutAll.

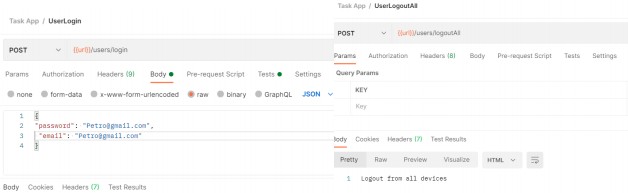


1. Створити обробник маршруту, що видаляє всі токени .
2. Надіслати 200 або 500.

app.post('/users/logout', auth, async (req, res) => {  
 try {  
 req.user.tokens = req.user.tokens.filter((token) => {  
 return token.token != req.token;  
 });  
 await req.user.save();  
 res.send();  
 } catch (e) {  
 res.status(500).send(e.message);  
 }  
});

1. Протестувати роботу: залогінітись декілька разів та здійснити logoutAll. Пе-ревірити базу даних.



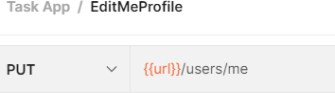




**Завдання 2.**

Здійсніть рефакторинг обробника для внесення змін для користувача.

1. Задати шлях /users/me.

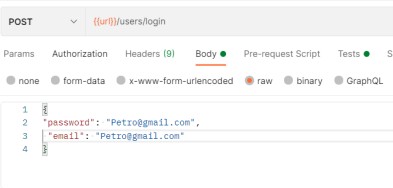


1. Додати middleware аутентифікації в обробник.

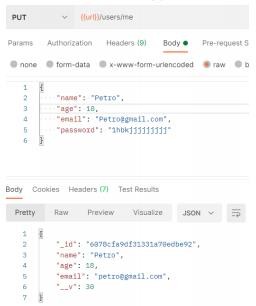
***router***.put('/users/me', auth, async (req, res) => {  
 try {  
 const updates = ['name', 'email', 'password','age'];  
 updates.forEach((update) => req.user[update] = req.body[update]);  
 await req.user.save();  
 res.send(req.user);  
 }catch (e) {  
 res.status(500).send(e);  
 }  
});

1. Використати існуючий запис користувача замість запиту по id.
2. Протестувати роботу через Postman: змінити вік, ім’я, пароль та спробувати здійснити вхід під старим паролем і під новим.

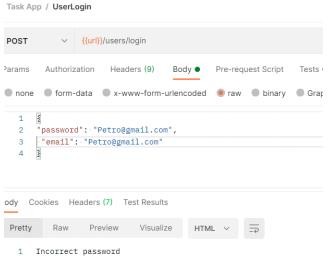
Виконуємо спочатку вхід



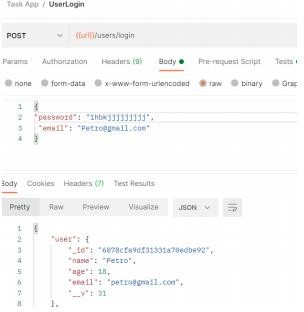
Змінюємо дані



Заходимо під старим паролем і отримуємо помилку



І успішно входимо під новим

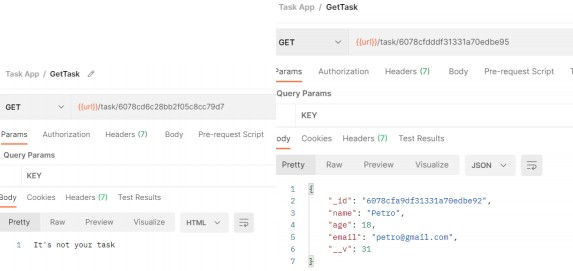


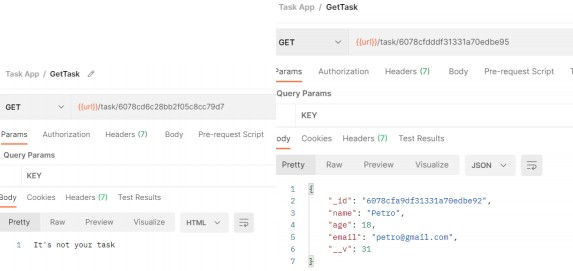
**Завдання 3.**

1. Реалізуйте отримання задачі по її id лише для користувача, якому належить да на задача. Для авторизованого користувача, якому не належить дана задача,

надіслати помилку 404.

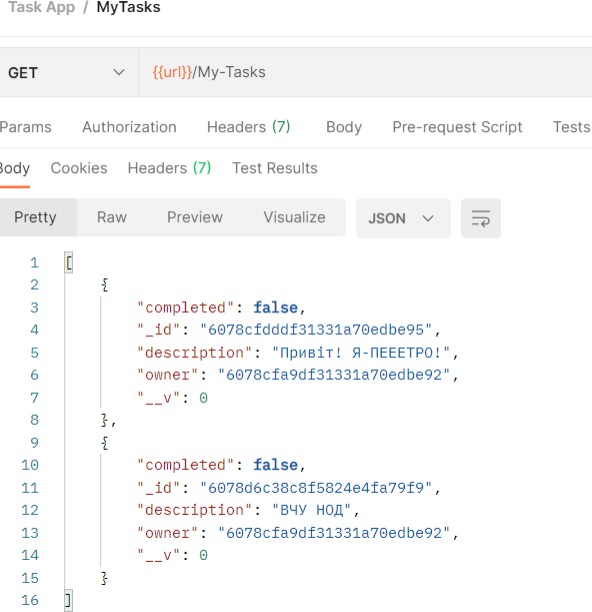
***router***.get('/task/:id',auth,async (req, res) => {  
 try  
 {  
 let taskId = req.params.id;  
 const task = await ***taskModel***.findById(taskId);  
 await task.populate('owner').execPopulate();  
 if (task.owner.id === req.user.id){  
 res.send(task.owner);  
 }else{  
 res.status(404).send("It's not your task")  
 }  
 } catch {  
 res.send({ code: 404, message: 'Not found' })  
 }  
  
  
})





2. Модифікуйте метод отримання списку задач, лише таких, що належать даному користувачу.

***router***.get("/my-tasks",auth,async (req,res)=>{  
 const user = req.user;  
 await user.populate('tasks').execPopulate();  
 res.send(user.tasks);  
})



3. Створіть метод редагування задачі лише тої, що належить даному користувачу

***router***.use("/task-update/:id", auth, async (req, res) =>{  
 let taskId = req.params.id;  
 const task = await ***taskModel***.findById(taskId);  
 await task.populate('owner').execPopulate();  
 if (task.owner.id === req.user.id){  
 const updates = ['description', 'completed'];  
 updates.forEach((update) => task[update] = req.body[update]);  
 await task.save();  
 res.send(task);  
 }else{  
 res.status(404).send("It's not your task")  
 }  
});

